
A software tool for simulating practical chemistry

David Povey and James Bennett
Department of Chemistry, University of Surrey
email: d.povey@surrey.ac.uk

A software package has been written to allow a user to build and manipulate a simple chemistry experiment. Using a toolbox of equipment the apparatus can be interactively designed and the necessary chemicals added from a database. Selection of the appropriate physical and reaction conditions allows the experiment to be run both in real and virtual time, snapshots of the experiment being stored for subsequent modification and replay. The structure of the reaction data file allows any reaction to be designed with yields and both forward and backward reaction rates. Thus, the user has the opportunity to experiment with the best apparatus layout, reactant composition and physical conditions in order to achieve an optimal result. Some extensions of the current software are discussed.

Introduction

The field of 'active learning' is growing rapidly in many areas and high-quality software that allows the student to interact with a particular environment is now becoming commonplace. Chemistry is no exception and one area that has seen extensive development is the use of computer simulations of laboratory procedures used either as a preparation for laboratory work or to carry out a virtual investigation. In the former case, Nicholls (1999) describes a range of software providing pre-laboratory support for a range of first-year undergraduate inorganic chemistry experiments. Using these computer programs, students work at their own rate on material presented through graphics, animations, simple calculations, tests and questions prior to carrying out the actual laboratory exercise. It is claimed that this approach encourages students to evaluate critically procedures and results by ensuring that they think beforehand about the tasks they will be performing subsequently in the actual laboratory class. Garratt (1997) describes a different approach, predominantly in the areas of physical chemistry and biochemistry, giving students the opportunity to design 'virtual investigations' and to process and interpret the resulting

data. For example, students are asked to choose a set of experimental parameters to allow the kinetics of a particular enzyme (randomly generated from 150,000 hypothetical examples) to be investigated.

The decision to develop the software described in this paper was in part a response to the results of a brief survey of our second-year undergraduates on the merits of practical courses they undertook. In the second year of our chemistry degree programmes, the laboratory practical course occupies one day per week for the whole academic year with equal time allocated to physical, inorganic and organic chemistry. In the physical chemistry component, the experiments involve using a range of techniques to make particular measurements on a system followed by a subsequent analysis of data. The inorganic and organic chemistry sessions are primarily preparative in nature, studying various practical techniques, although the materials synthesized might be analysed subsequently. Additionally, all students undertake a twelve-week course, 'Determination of Structure'. In this course, students working in pairs, are given a sample which they are required to identify based upon chemical and spectroscopic evidence. The evidence is provided in a folder of spectra and supplementary data, but students may make some limited experimental measurements.

Our survey elicited an interesting range of responses. The Determination of Structure course was reckoned to be 'stimulating and interesting' often representing in their eyes 'a thought-provoking exercise and the opportunity to develop problem-solving strategies'. Comments on our traditional practical courses were more negative. Those classes that involved synthesis were ranked generally between 'interesting the first time' to 'repetitive and boring'. When questioned further, having explained that learning about synthetic procedures was important training for practical chemistry the responses change to 'well, we're only following recipes; if we do this we know we'll get the correct answer, where's the challenge in that'. The physical chemistry did fare better. In this case, students saw the interpretation of the data as the challenge but the collection of data as 'following a recipe, no opportunity to experiment'.

The need to develop skills in the design and setting up of an experiment is crucial to the practical training of professional chemists. However, many practical courses are formulaic in approach requiring students to follow pre-planned recipes. While students gain some familiarity with chemical glassware, and instrumental techniques reinforce theoretical aspects of a course, opportunities for self-learning and experimentation in this type of environment can be difficult to implement when students follow the script like a cookbook (Johnstone, 1980). Safety considerations often play a part here, individual experiments needing to be risk-assessed before allowing them to be carried out. Of course, useful pre-lab work can still be carried out by allowing the student to view a video-based explanation of practical techniques (Rest, 1995). However, the rather sterile nature of the practical side of the experiment can still remain.

This in no way decries the importance of the practical. It is possible to design particular synthesis experiments which allow a certain freedom, a scenario which has been successfully implemented (Osborne, 1993). Here, the student might be asked to carry out the process using a range of different conditions and decide upon the best procedure. The instructions are clear, the recipe needs to be followed, but not now in a totally linear way. If

all the various combinations have been risk-assessed, the safety aspects have been covered. The only problem here may be one of time constraints where extended investigations may not fit into the available time slot.

It has been suggested (Verdonk, 1993) that the laboratory experience can be improved by turning the process into something that more closely resembles a true investigation. In this respect, an alternative approach can be to provide more of an open-ended investigation requiring the planning and carrying out of a synthesis. The planning aspects can be carried out 'dry' by reference to textbooks and journals. However, this is often no more than a literature search: it is the actual practical aspects and decision-making processes that are more difficult to deal with because of the inherent problems associated with risk-assessment of an individual experiment.

We have set out to investigate ways in which practical work might be supported by designing and developing a software tool to simulate basic bench chemistry. The use of the software is intended to augment laboratory sessions, not to replace them. As with a real practical experiment, the application should respond to what the user does and produce products and yields dependent upon the apparatus layout, reactant composition and specific physical conditions.

The development of the software consisted of four components:

- a mechanism for the user to design chemical apparatus and assemble it in a fashion similar to the well known and commonly used Quick Fit glassware;
- a means of selecting and adding specific chemicals to the assembled apparatus from a database;
- a simple mathematical model to simulate the performing of reactions on the reagents under a range of conditions;
- a visual interface to allow the reactions to be followed and replayed under different conditions so that various experimental scenarios may be investigated.

Design rationale

The program has been designed around a series of tasks, all controlled by a container application. Each component that requires information about the experiment shares a data class containing that information that can be passed easily between components. The components that need the experiment data are all implemented in the container application as a series of objects. Any components that do not require the experiment data are implemented as COM objects, using Microsoft's Component Object Model. This approach has been adopted to optimize data transfer speed. Borland Delphi 3.0 was chosen as the major development language. It is fully object orientated with extensive file streaming and database capabilities.

Application structure

The application itself consists of different components as shown in Figure 1. The container application is made up of a number of internal objects as shown in Figure 2 and is the main front-end to the application'

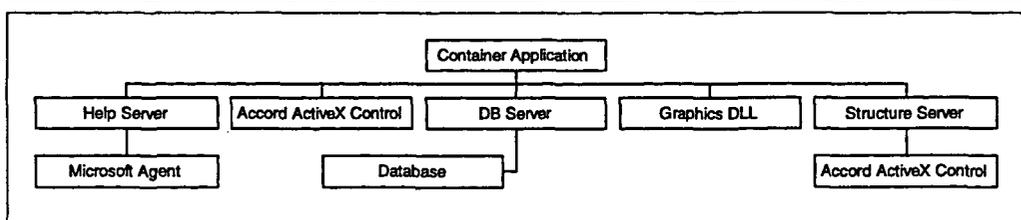


Figure 1: Object model of the software.

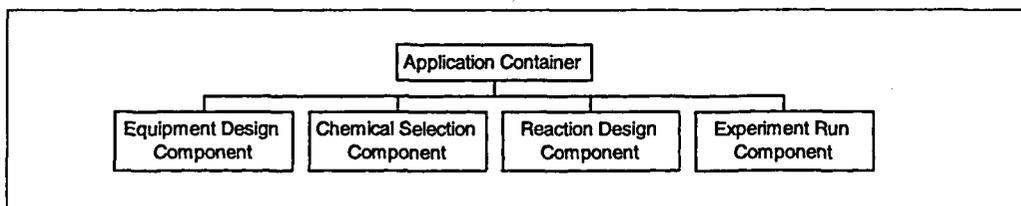


Figure 2: Components in the Container Application.

The main function of the Container Application is to display a diagram of the experiment and allow the user to run the other components in the system. It loads and saves experiments and any stored runs to and from files by writing the components to a file stream and runs the other components by passing the experiment to them. When the user selects to run the experiment, it checks if apparatus, chemicals and reactions have been set. If not, it flags this and informs the user, giving them the option to continue. It also provides printing functionality, and can either output a picture of the apparatus or an experimental report that has two sections, an experimental section and the results. The experimental section provides a picture of the experiment, a list of the apparatus including any relevant settings (e.g. heater temperature), a list of the chemicals, their temperature, state, quantity and the equipment they are in, and a list of the reactions flagged to take place in the experiment. The results section gives the final reaction time and the final products. It is thus possible for a student, as part of an assignment, to produce a complete printed record of the progress of the experiment. Also implemented is a Wizard that guides the user through the various stages for building and running an experiment. As described later, access to a spoken interface is possible.

Data class structure

The data class structure defined for use in the software is shown in Figure 3. The top level TExperiment contains a collection of equipment classes, a collection of reaction classes and an image of the experiment. Each equipment class contains a list of the connection points around and a list of the chemicals contained in it. Each chemical class has a structure stored as a MDL/MolFile, as well as physical properties such as specific heat capacity, melting point and boiling point. The reaction classes are a collection of different reactions, each with a list of starting materials, a list of products, the temperature of reaction and a description. Each starting material and product has a stoichiometry, which

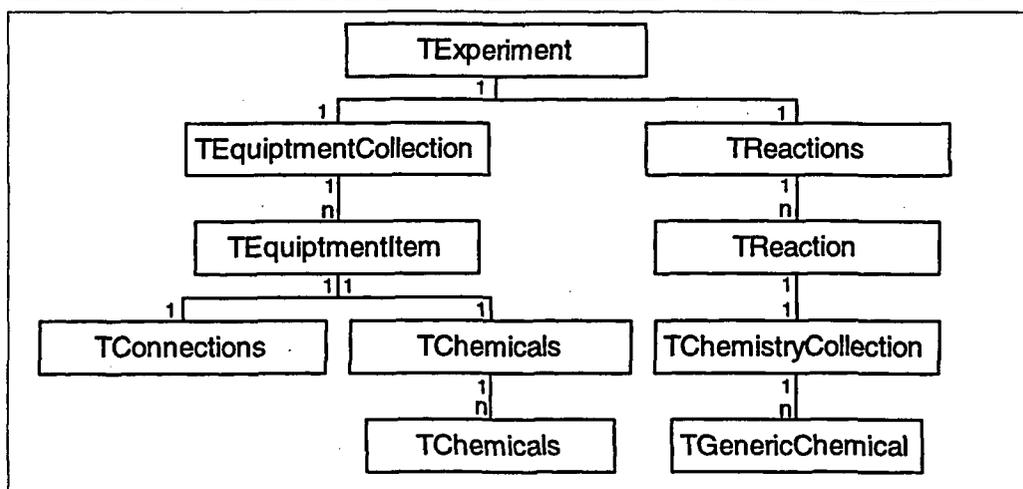


Figure 3: Class diagram of the data structure.

gives the ratio of the amount of starting materials that react and products that are produced. This class structure mirrors the hierarchy of a real experiment and provides a logical way to store all necessary data. These classes also contain methods to perform all manipulation, from adding and deleting chemicals to deciding where to place equipment during the experiment design process.

Equipment design component

This component allows the user to design and assemble the apparatus used in the experiment. A simple 'drag and drop' approach has been utilized to allow glassware to be selected. Figure 4 shows the pilot version of the equipment design interface.

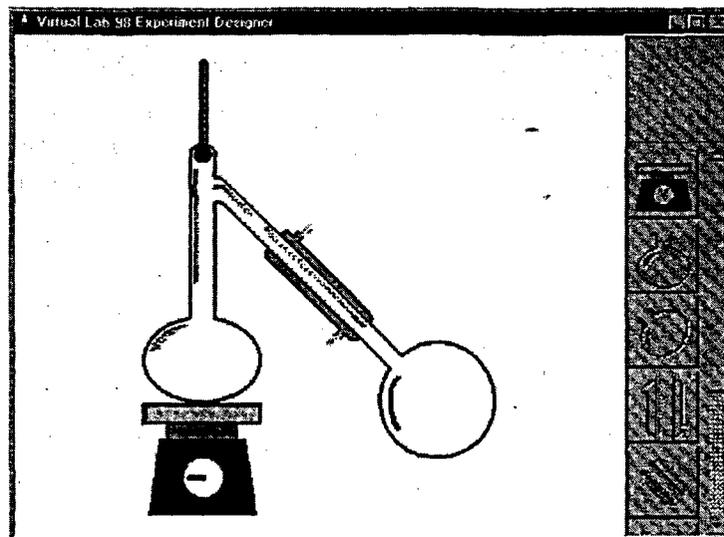


Figure 4: The equipment design interface.

Two strategies for assembling the equipment were considered. The first restricts the user's choice. In this case, the program would not allow an experiment to proceed unless the user had assembled the correct pieces of glassware in the correct positions with the correct chemical proportions and conditions set up. Then the application is run in a pre-set fashion allowing minimal or no interaction from the user. The second approach is a 'totally free form' where the user can place any piece of glassware anywhere on the screen adding chemicals where appropriate and setting reaction conditions accordingly. This allows the greatest flexibility and offers more learning potential as there may be several ways of achieving the desired product. However, this approach can lead to ridiculous and unrealistic scenarios. In this pilot version we have chosen to restrict the manner in which the apparatus can be assembled, but to allow free choice as to amounts and type of reagents. This avoids the specification of impossible scenarios while allowing some freedom of input.

The apparatus is stored in the experiment as a collection of pieces of equipment where each piece is a class with a number of properties; generally, a list of connections, a picture and a size. At present, for this pilot version, we have a limited range of glassware available but since an external file and its associated DLL are used to store this information, the equipment list is relatively easy to update. One can start at any point in building up the apparatus by selecting from the toolbox. However, once a piece of equipment has been selected, only those pieces that can logically connect to it remain enabled in the glassware toolbox. The equipment can also be saved at any time during the building process and reloaded as required.

Chemical selector component

This component is shown in Figure 5. It is used to design what reactions will take place in a particular experiment, or design new ones. The user can select what chemicals to add to the apparatus and in what quantity.

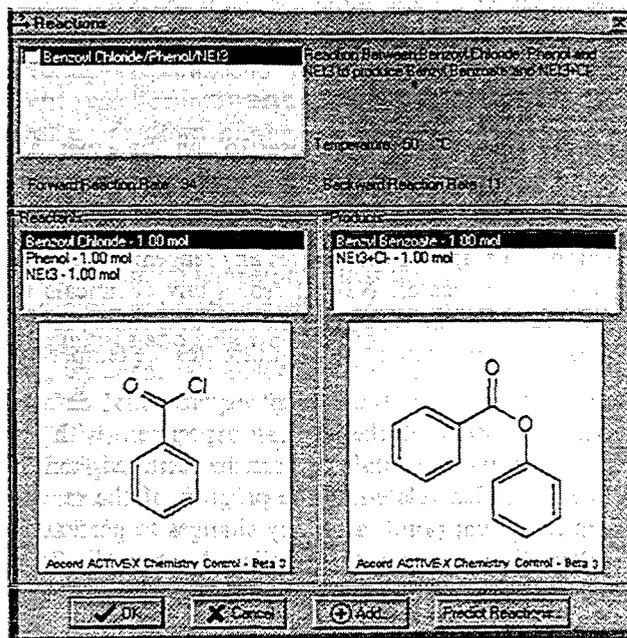


Figure 5: The reaction design interface.

A reaction is assumed to be composed of a number of reactants, a minimum temperature for reaction and forward and backward reaction rates. At present, there is no maximum temperature or any temperature dependence for the rates although future versions will have this implemented. Whilst this does represent a high level of approximation, reactions can reach equilibrium because both a forward and backward pseudo-rate is applied during the course of the reaction

By default, the chemicals are added at 20°C. Each reagent comes from a chemicals database, implemented as a *Paradox 5* table containing chemical structure and a range of physical properties such as molecular formula, molecular weight, specific heat, latent heat, melting and boiling points, density and name. Chemicals can also be added by providing the structure as a MDL/MolFile together with the other physical properties. The Accord ActiveX control (see later) with which we have experimented then calculates the molecular formula and molecular weight. There is no limit to the number of chemical entries in the database provided that the necessary physical properties can be allocated. Whenever a chemical is added to the apparatus, the amount required is requested and any units may be used as the conversion to moles takes place internally. The Chemical Selector displays the chemicals to be used in the experiment as a tree diagram with each chemical a child of the equipment it resides in.

The reactions are stored in a file that is loaded each time this component is accessed and can be edited as required to add or delete possible reactions. Any number of different reactions can be set, allowing the experiment to produce one product which can subsequently react again to produce another. A built-in utility is used to decide what reactions, both forward and backward, may take place using the chemicals already added to the experiment. This is done by matching the list of chemicals added to the apparatus with all of the reactions that have been set up and flagging those that are possible. In the final version of the software, it is expected that this component will be hidden from the user and it will be up to the instructor to decide the types of reactions that can take place and those that will be investigated.

Experiment run component

This component allows the user to run the experiment at half, normal (one update per second), double and quadruple speeds. The component is shown in Figure 6.

When the simulation runs, it updates the progress once every time slice in a sequential fashion. This update first moves any reagents, then heats up the heater and any reagents in flasks on the heater. Next it reacts any chemicals based on the physico-chemical data extracted from the reaction database and finally it removes any chemicals that have zero moles in the experiment and equilibrates the temperatures.

The software stores the current experimental data for each time slice and provides an animated display of the running experiment with the reagents in their correct physical states. The experimental data can be retrieved from any point during the experiment by moving a slider, related to the progress of the reaction, and the experiment can be run again from that point, after any changes to particular properties have been made. It also displays the experimental details as a tree diagram showing the equipment and any chemicals in specific parts of the apparatus, along with their temperature and physical state and the reactions that could take place.

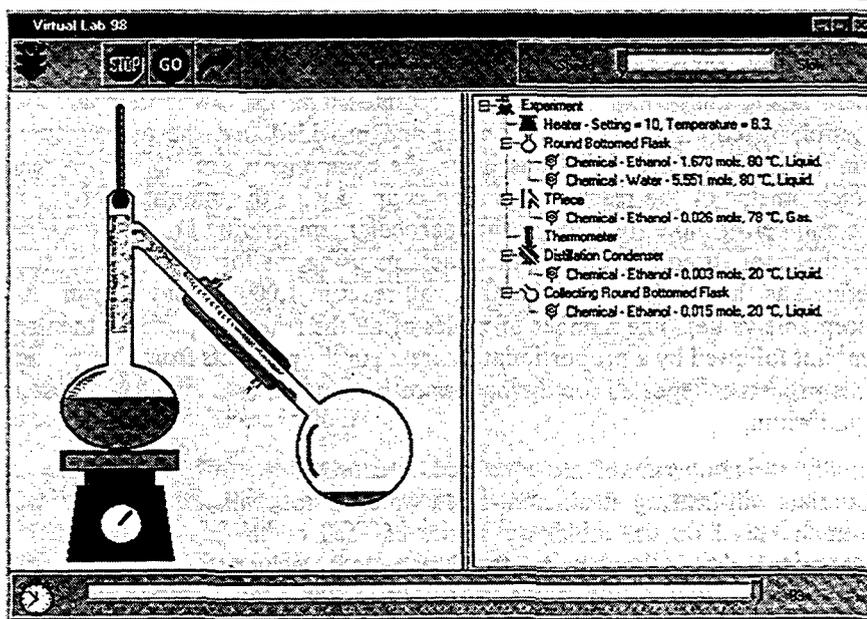


Figure 6: The experiment run component.

Because of the complexities of even a very simple system, a number of rules and simplifications have been implemented in this version. All chemicals are moved using gravity with the assumption that all gases are less dense than air. The experiment runner loops through the equipment looking for liquids and solids. If it finds any, it checks the equipment for any downward connections and if any are present, the reagents are moved downwards into whatever is below, either a piece of glassware or out into the surroundings if no glassware is present. This is repeated until no more movements can take place. Gases are then moved upwards by checking for upward connections and finding a logical path via those connections either to an adjacent vessel or into the open air. This is also repeated until no more gas moves. This logical up or down movement makes the movement of chemicals simple but does not at present allow for dense gases.

Providing energy to the system proved to be more problematical. The heaters used have a temperature setting from 1 to 10 indicating how much energy they put into the system in unit time and a heat setting, which is the maximum temperature that the system can reach. Thus, a low temperature setting and a high heat setting may mean the system will never reach equilibrium. This only applies to liquids and solids below their boiling point. If the temperature exceeds the melting point, the chemicals melt completely and immediately. If the temperature reaches boiling point, they are boiled slowly (the display changing to show this effect visually). This is achieved by utilizing another property, the latent heat of evaporation. This works in an identical way to the specific heat capacity, except every time the value is exceeded, one gram (a reasonable but arbitrary quantity) of the chemical is converted to a gas. These methods of heating the chemicals and converting the state provide a good method of qualitatively showing the reaction as it happens. Reactions happen fast enough to occur in a few minutes, but not so fast that the user cannot witness them happening.

Once the chemicals have been heated in that cycle, a check is made for any possible reactions. This is done by looping through the equipment, and assembling a list of chemicals in each component of the apparatus. This list is then compared with the list of possible reactions that can take place and provided that the temperature is above the minimum set in the reaction designer, the components are allowed to react. Using the stoichiometry of the reaction equation coupled with the quantities of the reagents used, the number of moles reacting at that particular temperature in that time slice is calculated for the smallest component of the reactants using the forward reaction rate. This rate is defined as the number of moles that will react in 1,000 seconds (again an arbitrary but acceptable value). The number of moles of all the reacting species is then reduced by this amount followed by a proportionate increase in the products from that particular reaction. This process is repeated for the backward reaction, thus allowing the system to come to equilibrium.

Finally, the temperatures are equalized. Because of the different specific heat capacities, materials will heat up at different rates, which is not realistic. Therefore, we use a simple formula based on the molar quantities of each of the reactants to define an average temperature that is then applied to all reagents in each specific piece of equipment. While not providing completely accurate temperatures, it has the advantage of speed and simplicity.

Interface components

In this section we discuss the help facilities provided by the software and the techniques adopted for displaying and manipulating chemical formulae.

Help server

The Microsoft Agent (available free from <http://www.microsoft.com/MSAgent>) is a set of components that will be an integrated part of Windows 2000. It is a tool designed to improve user interaction by providing an animated character that communicates with the user at three levels, depending upon what additional components are installed. These levels of communication are speech bubbles containing text, spoken output using a text to speech engine, and vocal control, using a speech recognition engine. For the pilot version, we experimented with speech bubbles and spoken output. We implemented a conventional text-based 'wizard' for aiding the user in the design, building and running of experiments and then refined the user interaction with the Microsoft Agent. There was no doubt that it improved substantially the impact of the interface despite a slow response.

Displaying and manipulating chemical formulae

During the design stage, it was decided that the program should be 'chemically aware' so that all chemical structures were stored in a standard format that could be analysed or modified by any additional chemistry software. Thus, we have experimented with a collection of development tools, the Accord Suite, which provides functions for creating, manipulating and performing database functions on chemical data, and an ActiveX control for displaying and manipulating chemical data in an application or on a Web page. The early release/beta test version of the ActiveX control was used (available free from <http://www.synopsys.co.uk/>). This ActiveX control allows chemical information to be imported from a file, either locally or from an internet site, and many standard file types

are supported including Daylight SMILES, MDL/MolFiles, ISIS/Draw sketches and ChemWindows sketches. The control also has functions for obtaining information about the molecule stored in it, including its molecular weight and empirical formula.

The current implementation of the Accord ActiveX control allows structure to be visualized and some molecular properties to be calculated. The Accord suite also includes a chemistry Application Programmer Interface (API) that could be implemented. This API provides functions for comparing structures against generic structures, and producing products from generic structures and starting materials.

Conclusions

During the development of the program, it became clear a large number of possible enhancements could be made. These include:

- While two-dimensional flat representation for the glassware was adequate, the ability to manipulate apparatus showing a true three-dimensional perspective may be advantageous. Dolata and Wathen (1998) have reviewed the use of Virtual Reality Modelling Language in chemical education and have shown examples of chemical apparatus containing such a perspective although no interactive design capability was included.
- There are other operations that could be carried out such as allowing the user to change more than just the heater and funnel properties during a reaction run, e.g. refilling a funnel or allowing water to flow through a condenser. In fact all the manual operations might be required.
- The current prototype does not implement a true kinetic model where rates of reactions are dependent upon temperature. Whilst it is not always possible to acquire correct kinetic data for reactions, Povey and Baldwin (1997) as part of the eLABorate project have developed a method which allows the kinetics of any reaction to be simulated given the order, activation energy and Arrhenius factor. This means that the kinetics (which need not necessarily reflect the true scenario) can be built into the software increasing the range of opportunities for investigation.
- Currently all reactions/reagents are set to a starting temperature of 20°C and return to this temperature upon cooling. This is very restrictive as many reactions require the presence of an ice-bath so a facility to allow a greater range of operating temperatures could be implemented.
- At present any products from one reaction cannot be stored and used as starting materials for another. Many syntheses require a multi-stage pathway and it is clear that a varied and complex range of scenarios could be developed if this was allowed.
- Currently, solutions are not modelled as this would require the database to hold information on solubility data in a range of solvents. However, if implemented, this would increase the realism of the scenario allowing a user to select the best solvents for a reaction and simulate precipitation and recrystallization events.

The software is certainly not meant to replace traditional practical chemistry. Rather it should be seen as an adjunct in which students can experiment, in their own time, with a

range of scenarios and attempt to design a strategy to optimize a particular synthesis. There are no time constraints, so poor yields can be discarded, the experiment replayed and conditions or reagents modified. The range of scenarios is unlimited, provided that the necessary reactions can be coded into the database and with some prior thought, the lecturer can place specific restrictions on any task in terms of yield, number of steps, types of reagent used and so on.

References

- Dolata, D. P. and Wathen, S. P. (1998), 'The use of VRML in chemical education', *Journal of Molecular Graphics and Modelling*, 16, 37-45.
- Nicholls, B. S. (1999), 'Pre-laboratory support using dedicated software', *University Chemistry Education*, 3, 22-7.
- Garratt, J. (1997), 'Virtual investigations: ways to accelerate experience', *University Chemistry Education*, 1, 19-27.
- Johnstone, A. H. (1980), 'Chemical education research: facts, findings and consequences', *Chemical Society Reviews*, 9, 365-90.
- Osborne, A. G. (1993), 'The open-ended experiment - more than a practical', *Proceedings of Variety in Chemistry Teaching Conference*, University of York, UK, 78.
- Povey, D. C and Baldwin, C. R. (1997), 'Program kLAB',
<http://www.york.ac.uk/depts/chem/staff/lelaboratel>.
- Rest, A. J (ed). (1995), *Basic Laboratory Chemistry*, Harrow, Middlesex: Educational Media Film and Video Ltd.
- Verdonk, A. (1993) 'The role of educational research in the quest for quality in the teaching and learning of chemistry', *Proceedings of Variety in Chemistry Teaching*, University of York, UK, 49-54.